III. REMARKS

1. Claims 1, 9, and 12 are amended. The 35 U.S.C. §112 rejection is addressed.

2. It is submitted that Stern does not anticipate claims 1, 5, 8, 9, 11 and 12 under 35 U.S.C. §102(e).

Claim 1 recites data transfer means for downloading a driver from a network server, an interface unit configured to transmit and receive data to and from an external unit for processing data provided in a specified format, transmit a signal indicative of attachment of the external unit to the network element, receive an address for the downloading of said driver in response to said signal from the external unit and provide an address from which the driver is to be downloaded.

The interface unit obtains the address for the downloading of the driver in response to the sending of a signal indicative of attachment of the external unit to the network element. Claim 1 also recites verifying an origin of the downloaded driver.

At least these features are not disclosed or suggested by Stern.

Stern is directed toward embedding a secure network management function in a network interface device. (Col. 3, lines 40-42). The network interface is referred to as a "Java enabled Network Interface Device". "Java" is a general purpose, object oriented programming language that allows the secure delivery of software components. (Col. 5, lines 28-40). Stern discloses a Java Enabled Network Interface device. The interface device is provided in association with a host computer. The interface device includes a processor, and a volatile and non-volatile memory. The interface device acts a proxy between the host computer and a remote server. A Java Virtual Machine is loaded in the interface device. The Java virtual machine is loaded securely and the boot loader verifies that the virtual machine has not been tampered with. The verification is performed using a digital signature of the virtual machine provider. The virtual machine controls access to network resources within the interface device. These resources

include, for example, buffer memory, routing tables or packet scheduling processes. Sole access to items of security such as the network resources or token objects is through Java applets. The digital signature of the Java applets is verified before allowing them to execute commands that alter the state of the items of security. In case the remote server wishes to give commands to the interface device, such as to alter the QoS admitted to the host, the commands must be provided with a valid digital signature. The Java virtual machine verifies the digital signature before allowing the execution of an applet, which in turn executes the command.

In Stern, the Java Virtual Machine 402 processes all data between the host computer 405 and network 440. (Col. 7, lines 16-20). (See FIG. 5). The incorporation of the Java Virtual Machine in the network interface device allows the device to store state information. Thus, the host computer can access state information from this interface device instead of over a network from a remote server. (Col. 8, lines 44-57). The Java Network Device of Stern is a proxy for network services. (Col. 8, lines 57-59) and Stern stores an object of value in the network interface device. (Col. 9, lines 3-7).

During operation, the Java Network Device of Stern monitors inbound and outbound traffic between a computer 610 and processor 620. (Col. 9, lines 48-53). Traffic directed to a Java applet can be detected. (Col. 9, lines 54-57). The Java network device can also intercept outbound traffic 611 and respond. (Col. 10, lines 3-7).

To install an object of value in the Java Network Device of Stern, a Java applet is loaded. (Col. 10, lines 14-22). The applet is loaded directly from the network correction 722. (Col. 10, lines 20-22). The object of value is initially available from a license process 714 within a server application 712 running on a secure server 710. (Col. 10, lines 22-26).

The interface device of Stern may act as a temporary storage or local token dispenser for token objects on behalf of the remote server, when the remote server is not available. The token objects can represent charging credit values or software licenses. The interface device intercepts communications addressed to the remote server by using

an IP address and a port number associated with the remote server as a filter. The IP address and the port number are provided as the applet and the token object is provided for the first time from a remote server to be cached in the interface device. (Col. 10, lines 29-48.) When the host computer requests a token object from a remote server, the remote server first responds with an applet representing the token object with an initial value. The applet is routed from the remote server to an address of the interface device and the virtual machine therein. After providing the applet, the remote server provides a reply to the address and the port number from which the request for a token object originally came. The request application knows that the request has been complied with and may continue execution. (Col. 11, lines 13-31). The interface device and the virtual machine intercept subsequent requests pertaining to the token object to be processed by the applet.

Stern does not disclose or suggest downloading a "driver" from a network server, where an "address" of the network server is provided by an interface unit. Stern does not disclose or suggest transmitting a signal indicative of attachment of an external unit to a network element and receiving an address for the downloading of said driver in response to said signal from the external unit. Stern merely teaches that an "object of value" is installed.

In this regard, the local application 704 of the host computer 702 requests a "token" from the application 712 or the server 710. In response, the server 710 sends an "applet" representing the "object of value" to the Java Network Device 706 and includes the IP address of the host computer. The server 710 returns a remote procedure call to the host 702. When the local application 704 on the host 702 requests the "object of value", the request is "intercepted" by the Java Enabled Network Interface Device 706. (Col. 11, lines 13-31). The "object of value" is then processed. (Col. 11, lines 32-46). However, there is nothing in Stern to suggest downloading a driver, where the address of the server from which the driver should be downloaded in provided by an interface unit. The "device" the examiner refers to is the "application 704" of host 702. The "request" may include the IP address of "host computer" 702. (Col. 8, lines 21-24). This

is not the same as the address of the server from which the "driver" is to be downloaded. Thus, applicant respectfully traverses the Examiner's use of the term "inherently" in this respect, and requests evidentiary proof that Stern discloses that the network "address" from which the driver is to be downloaded is taught by Stern.

Stern also does not disclose or suggest that the remote server provides an address from which a driver, the applet, may be downloaded to control its function. Rather, all that Stern discloses is that an applet is obtained as a by-product of requesting a token object. It will be appreciated that the address for the token object will be obtained from another source, which references the remote server and the token object therein. For example, a software installation disk or a game program downloaded from another remote host. Thus, Stern does not disclose or suggest each feature of claim 1, and the claim cannot be anticipated.

Claims 9 and 12 recite similar features that are not disclosed or suggested. Claim 5, 8 and 11 should be allowable at least by reason of their respective dependencies.

3. Claims 2, 6, 7 and 10 are not unpatentable over Stern under 35 U.S.C. §103(a) at least by reason of their respective dependencies.

4. Claims 17 and 18 are not unpatentable over Stern and Montgomery et al. ("Montgomery") under 35 U.S.C. §103(a) at least be reason of their respective dependencies.

Furthermore, there is no motivation to combine Montgomery with Stern. "Motivation" for purposes of 35 U.S.C. §103(a), requires some teaching or suggestion in the references themselves to make the proposed combination. It is submitted that there is no such teaching.

Stern is directed to embedding a network management function in a network interface device. (Abstract). FIG. 4 illustrates the architecture of the arrangement. Montgomery

is directed to smart cards. However, it is respectfully traversed that a "smart card" is not a "computer" as stated by the Examiner.

Rather, in Montgomery, a "smart card" is connected to a terminal. The "terminal" may be connected to a "host computer". The "smart card" can "access" the resources connected to the terminal, the host computer, and the network. (Col. 1, lines 55- Col. 2, line 2). The smart card can also "store" programming that is executed by the microcontroller, which initiates communication with the terminal. (Col. 2, lines 3-13). The "smart card" may "access" the resources "connected to" the terminal, the host computer, and the network. (Col. 2, lines 9-13). The host may be connected to remote hosts via a network such as the Internet. The communication between the smart card and the host is in full duplex. The smart card may initiate communications independently and issue instructions to the host computer. Thus, the "smart card" could not replace the host computer of Stern as is suggested.

Additionally, Stern and Montgomery are "non-analogous" art for purposes of 35 U.S.C. §103(a). References may be combined under 35 U.S.C. §103(a) only if the references are analogous art. In this case Stern and Montgomery are not analogous art. A reference is analogous art if:

1) The reference is in the same field of endeavor as the applicant's, or

2) The reference is reasonably pertinent to the particular problem with which the applicant was concerned.

Stern and Montgomery are not in the same field of endeavor. Stern is directed to a Java Enabled Network Interface device, while Montgomery is directed to smart card systems.

Applicant's claims are directed to supporting applications for network elements. Neither Stern nor Montgomery address this problem.

Thus, Stern and Montgomery are not analogous art, and cannot be used to establish a *prima facie* case of obviousness under 35 U.S.C. §103(a).

For all of the foregoing reasons, it is respectfully submitted that all of the claims now present in the application are clearly novel and patentable over the prior art of record, and are in proper form for allowance. Accordingly, favorable reconsideration and allowance is respectfully requested. Should any unresolved issues remain, the Examiner is invited to call Applicants' attorney at the telephone number indicated below.

The Commissioner is hereby authorized to charge the fee for a three-month extension of time ($1,020) and any other fees associated with this communication or credit any over payment to Deposit Account No. 16-1350.

Respectfully submitted,

Geza C. Ziegler, Jr.
Reg. No. 44,004

28 AUGUST 2006
Date

Perman & Green, LLP
425 Post Road
Fairfield, CT 06824
(203) 259-1800 Ext. 134
Customer No.: 2512

CERTIFICATE OF ELECTRONIC FILING

I hereby certify that this correspondence is being filed electronically addressed to the Commissioner of Patents, MAIL STOP AMENDMENT, P.O. Box 1450, Alexandria, VA 22313-1450.

Date: 28 August 2006

Signature:

Lisa Winefield
Person Making Deposit